WHAT IS CLAIMED IS:

1. A programmable finite state machine configured to transition to one of a plurality of next states from a current state in response to receipt of an input symbol, each of the current state and next states being represented by m bits and each input symbol being represented by k bits, the programmable finite state machine comprising:

a first memory configured to store a plurality of transition rules, said first memory further configured to receive a (k+m)-bit word representative of the input symbol and the current state and to supply one or more matching transition rules in response, wherein the one ore more matching transition rules are stored in a ranking order of generality;

a logic block configured to select a most specific transition rule from among the one or more matching transition rules; and

a second memory configured to receive the selected transition rule and to supply one of the plurality of next states in response.

2. The programmable finite state machine of claim 1 wherein the first memory is a ternary content addressable memory (TCAM).

3. The programmable finite state machine of claim 1 wherein the second memory is a static random access memory (SRAM).

4. The programmable finite state machine of claim 1 wherein the ranking order of generality is an increasing order of generality.

5. The programmable finite state machine of claim 1 wherein the ranking order of generality is a decreasing order of generality.

6. The programmable finite state machine of claim 1 further comprising:

a register configured to receive the k-bit input symbol and the m-bit current state and to generate the (k+m)-bit word supplied to the TCAM.

7. The programmable finite state machine of claim 1 further comprising:

a look-up table configured to receive the m-bit next state from the second memory and to generate an associated output.

8. The programmable finite state machine of claim 7 wherein the LUT contains a bit configured to accept or reject a request.

9. The programmable finite state machine of claim 1 wherein the logic block is a priority encoder.

10. The programmable finite state machine of claim 1 wherein the logic block is a priority arbiter.

11. The programmable finite state machine of claim 2 further comprising:

cascading a second TCAM to the first TCAM to increase a depth of the programmable finite state machine.

12. The programmable finite state machine of claim 2 further comprising:

cascading a second TCAM to the first TCAM to increase a width of the programmable finite state machine.

13. The programmable finite state machine of claim 1 where the input symbol is received from a digital network, and the programmable finite state machine is configured to provide signature-based network security services selected from a group consisting of network intrusion detection, network monitoring and surveillance, virus protection, traffic filtering, content and copyright classification, storage area networks, policy and access control auditing, spam detection and prevention, fraud detection and network forensics, content-aware switching, and message classification.

14. A method for transitioning to one of a plurality of next states from a current state in response to receipt of an input symbol, each of the current and next states being represented by m bits and each input symbol being represented by k bits, method comprising:

receiving a (k+m)-bit word representative of the k-bit input symbol and the m-bit current state;

supplying one or more matching transition rules, wherein the one or more matching transition rules are stored in a ranking order of generality;

selecting a most specific transition rule from among the one or more supplied matching transition rules; and

supplying the one of the plurality of next states.

15. The method of claim 14 wherein the one or more matching transition rules is supplied by a ternary content addressable memory (TCAM).

16. The method of claim 14 wherein the one of the plurality of next states is supplied by a static random access memory (SRAM).

17. The method of claim 14 wherein the ranking order of generality is an increasing order of generality.

18. The method of claim 14 wherein the ranking order of generality is a decreasing order of generality.

19. The method of claim 14 further comprising:
storing the k-bit input symbol and the m-bit current state.

20. The method of claim 14 further comprising:
generate an output for the one of the plurality of next states.

21. The method of claim 14 wherein most specific transition rule is selected by a priority encoder.

22. The method of claim 14 wherein most specific transition rule is selected by a priority arbiter.

23. The method of claim 15 further comprising cascading a second TCAM to the first TCAM to increase a depth of the plurality of next states.

24. The method of claim 15 further comprising cascading a second TCAM to the first TCAM to increase a width of the plurality of next states.

25. The method of claim 14 wherein transitioning to one of the plurality of next states from the current state is made in connection with a signature-based network security services selected from a group consisting of network intrusion detection, network monitoring and surveillance, virus protection, traffic filtering, content and copyright classification, storage area networks, policy and access control auditing, spam detection and prevention, fraud detection and network forensics, content-aware switching, and message classification.

26. A method of generating state transition rules, the method comprising:

forming a plurality of Boolean logic functions each associated with a different one of a plurality of next states to which transitions are configured to occur;

performing logic minimization operation on number of minterms associated with each of the formed plurality of Boolean logic functions;

identifying at least one of the plurality of Boolean logic functions having a smallest number of minimized minterms;

storing the minimized minterms of the identified at least one of the plurality of Boolean logic functions;

adding expanded minterms corresponding to minimized minterms of the identified at least one of the plurality of Boolean logic functions to the remaining ones of the plurality of Boolean logic functions;

performing a second logic minimization operation on number of minterms associated with each of the remaining ones of the plurality of Boolean logic functions;

identifying at least another one of the plurality of Boolean logic functions having a smallest number of minimized minterms following the second logic minimization operation; and

storing the minimized minterms of the identified at least another one of the plurality of Boolean logic functions; wherein the stored minterms define the state transition rules and are in a ranking order of generality.

27. The method of claim 26 wherein the minimized terms are stored in a ternary content addressable memory.

28. The method of claim 26 wherein the ranking order of generality is an increasing order of generality.

29. The method of claim 26 wherein the ranking order of generality is a decreasing order of generality.

30. The method of claim 26 further comprising:

storing a corresponding entry for each of the plurality of next states.

31. The method of claim 30 wherein the corresponding entry for each of the plurality of next states is stored in a static random access memory.